New Tricks for Old Data Sources: Mashups, Visualizations, & Questions Your ILS Has Been Afraid to Answer

Brian Norberg
North Carolina State University, brian_norberg@ncsu.edu

Darby Orcutt
North Carolina State University, darby_orcutt@ncsu.edu

John Vickery
North Carolina State University, john_vickery@ncsu.edu

Brian Norberg, Darby Orcutt, and John Vickery, "New Tricks for Old Data Sources: Mashups, Visualizations, & Questions Your ILS Has Been Afraid to Answer" (2011). *Proceedings of the Charleston Library Conference.*
http://dx.doi.org/10.5703/1288284314899

New Tricks for Old Data Sources: Mashups, Visualizations, & Questions Your ILS Has Been Afraid to Answer

Brian Norberg, Libraries Fellow, North Carolina State University Libraries
Darby Orcutt, Assistant Head, North Carolina State University Libraries
John Vickery, Collection Manager for Management & Social Studies, North Carolina State University Libraries

Abstract:
Libraries have more and better data than we do the means to readily make use of it. Collection managers at the NCSU Libraries face this challenge head-on by developing and guiding the development of new digital tools to more efficiently and fully address the real-life challenges of managing library collections, including physical space and move planning for a new library and storage facility; blending ILS, ILL, consortial holdings, and e-book data for highly contextualized views on local needs versus access; and harnessing the full text of dissertations in our Institutional Repository to visualize the interests and information needs of NC State researchers. While these projects have entailed SAS or Ruby on Rails programming skills and/or other "techie" tools, they are not beyond the capacities of collections and acquisitions librarians to lead or even fully carry out. Of utmost importance are understanding the capabilities of the digital tools and of developing clear, nuanced, and effective ways to apply their results.

The Collection Management Department of The NCSU Libraries has long emphasized the strategic use of data to support decisions and communication concerning our collections. In 2002, we were perhaps the first academic research library to create a position of "Collection Manager for Data Analysis," in order to purposefully focus human resources effort on learning and crafting new tools for data-driven collections work. Over many years, we have also drawn on the skills and ingenuity of participants in our highly selective NCSU Libraries Fellows program (http://www.lib.ncsu.edu/fellows) to bring fresh knowledge and new ideas to us; in fact, this panel is comprised of one current and two former Fellows. Now, nearly a decade later, the Collection Management Department has thoroughly integrated into our institutional culture a sort of entrepreneurial spirit in working with data, with all members of our department engaged in contributing to data analysis work, keeping up with the latest trends and software, or even creating our own tools for data-driven projects. We'll showcase just a few examples of this below.

We very specifically used the modifier "strategic" above when referring to our use of data because we want to make clear our approach to data. Libraries collect and have access these days to lots and lots of data, more than we perhaps could ever hope to analyze. Yet, much of it serves no more purpose than white noise, and so we seek to choose carefully to invest our scarce time in projects with the potential to provide significant and actionable results, as well as projects that will themselves save precious staff time going forward. Furthermore, we are not looking for silver bullets or magic formulae. We do not seek to abrogate our decision-making responsibilities but rather to hone and inform our skills in this area. The precise tasks and work of the collection manager are certainly changing dramatically, but the significant need for the professional judgment of the collections librarian has perhaps never been greater than in today's research library.

Below, we spotlight three projects exemplary of our use of advanced analytical tools for data analysis. None of the tools below lies beyond the reach of the average collection development librarian, or at the very least, of the average collection development librarian in partnership with competent IT support. Our hope is that these may serve as models for many of you, giving you ways of conceiving of local projects and of clearly communicating with more technically proficient partners in developing new tools or applications.

Using SAS® for Physical Collections Information
At NCSU Libraries, we are addressing the demands of managing and analyzing our collection head on by applying advanced analytical tools. One of the tools that we have begun to use recently is SAS. As the requirements of collection management at the NCSU Libraries have shifted more and more toward need for larger scale analysis, we have outgrown the tradi-

DOI: http://dx.doi.org/10.5703/1288284314899

tional tools. While advanced analytical tools such as SAS programming have a relatively steep initial learning curve, the payoff is worth the time investment. Using SAS, we are able to quickly, flexibly, and precisely answer questions about our collections that are difficult if not impossible to answer using built-in ILS reporting features or spreadsheets.

This section of our presentation will focus on what questions we needed to answer about our collections and why SAS is an excellent tool for the job. We will also show examples of the process from our ongoing project to plan for a new library.

What questions has our ILS been afraid to answer?
At the NCSU Libraries, our ILS can do many things but there is much more we can understand and see about our collections than the ILS will do. There just isn't a built-in button or report that can handle the "what-if" questions required of moving our collections to a new library with an automated retrieval system. Nor do the built-in ILS reports do a particularly good job at helping us to understand how the collection is used.

Here are a few questions relating to planning a new library, collection usage and consortial collecting that we have recently asked of our collections data.

Planning for the new library:

- "How many items of each type from each library location would be moved to the new library based on the following criteria:
    - Only items from the main library in 32 specific call number ranges
    - All items from one branch but none from the others
    - All government documents
    - All items in off-site shelving except "special collections type" items
    - No items that used to be at the Design branch library but were moved to other locations at some point in the past (by the way, our ILS doesn't track historical location)."
- "Based on all the criteria above, can you estimate how many new items we'll add each year to the new library? How many of these would go onto open shelving and how many would go into the automated retrieval system?"
- "How big of an impact will temporary closure of a subset of the collection have on faculty, grad students and undergrads?"
- "Should we move items in the Z class to the new library? What exactly do we have in the Z class anyway?"

How have our collections been used:

- "What percentage of items in the collection has circulated based on when they were added? What does the trend look like at the LC subclass level for each library location?"
- "Based on our print collection usage, how long should we leave PDA/DDA records available in our catalog?"

Consortial collecting analysis:

- "How much has the TRLN consortium duplicated print orders in the last four years—and in what subject clusters?"
- "Which publishers account for the greatest share of duplicate and triplicate orders across the consortium—and in what subject areas?"

Common to each of these questions are added layers of criteria and detail. While our ILS has built-in reports for summary and count statistics, each additional layer of complexity to a question makes the ILS reports less precise and useful. To get around the limitations of built-in ILS reports and do further analysis, collections librarians often turn to Excel spreadsheets or occasionally an Access database. Both Excel and Access have limitations that make using an advanced analytical tool a better choice.

In the following sections, we will look at what SAS is and why it is a better tool for large scale collection analysis.

What is SAS?
SAS is a licensed, integrated system of software for business intelligence, forecasting, data warehousing, statistical analysis and many more specialized functions. SAS is typically a programming based system. That is, programs are written to import, manage and analyze data. While SAS is often thought of as a statistics package, it is much more flexible and

robust. SAS also provides a graphical user interface which allows a point-and-click environment to the programs running in the background. New users can even access the generated code to begin learning SAS programming.

SAS consists of many separate, integrable components which are licensed and installed based on an organization's needs. Fortunately, in the academic environment, SAS is often licensed at the university level similar to other software products and may not even involve an additional cost to the library.

The following SAS component products were used in the analysis and planning projects that are underway at the NCSU Libraries. While each component adds significant value, it would be possible to do nearly everything using Base SAS alone.

- Base SAS: This is the core product of SAS. It provides the SAS programming language to perform data manipulation, analysis and reporting. Base SAS also integrates SQL.
- SAS/ACCESS: This allows seamless connections to third party databases as well as PC file formats such as Excel.
- SAS/GRAPH: This extends the graphing capabilities of Base SAS. The most current version of Base SAS (version 9.3), however, includes robust statistical graphing procedures.
- SAS Enterprise Guide: This is a point-and-click graphical user interface to underlying SAS programs. It can make the more complex aspects of graphing and reporting easier to use. Users can also access the underlying code to learn the SAS programming language.

Why use SAS

In a general sense, the first and foremost reason to use SAS is that it is a tool and programming language specifically designed for data analysis and manipulation. More specifically, there are several reasons why SAS is an excellent tool to apply to collections planning and analysis. Here we look at five of these reasons.

#1 SAS puts you in control of how you work with your data. The programming language is flexible with concise procedures that encapsulate analytical functionality. The data manipulation functionality allows you to transform, recode and prepare your data for analysis.

Any data analysis project often generates additional "what-if" questions. Because SAS is programming based (or stored processes based if you use Enterprise Guide), you can easily adjust and repurpose your programs to re-analyze your data in new ways.

While there is a steeper learning curve to SAS than with tools such as Excel and Access, the payoff is worth the investment. Learning just a few data manipulation (DATA Step) techniques and a couple procedures (PROC Step) can allow you to ask and answer questions of your collections data that were out of reach with typical tools.

#2 SAS can read data from any source. The SAS programming language allows you to import data regardless of format. This includes the raw output from your ILS ranging from the bibliographic tables to the transaction logs. SAS can also easily read data from Excel allowing you to synthesize the analysis that is stored away in all those spreadsheets.

#3 Big data is no problem. Our typical tools, Excel and Access begin to show serious strain in terms of performance and capability after a certain threshold. With SAS, we regularly analyze over 3.3 million bibliographic records—essentially our entire holdings. This allows us to analyze our entire collection population rather than using samples. Given the skewed nature of collections data, this can make our analyses much more precise and lead to better planning.

#4 SAS has robust and customizable reporting functionality. SAS can generate reports in many different formats and styles such as HTML, PDF, XML for Excel and RTF for Word. Multiple types of output can also be generated simultaneously from the same analysis meaning you can target more than one audience at a time.

The flexible reporting feature of SAS also extends the "what-if" capabilities of your analysis as it removes limitations to how data is presented. You are not locked into an application specific view of your data.

#5 Visualization functionality. Seeing is believing and by learning a few statistical graphics procedures you can turn your data into high quality, easily interpretable charts and graphs.

Why not use the built-in ILS tools

With our ILS, and perhaps most ILS, we are limited to a set of canned reports applicable to collections analysis and planning. These canned reports do not allow for easy customization or repurposing of the results and the output is limited to text files. Any customization requires the time of the ILS administrator who is busy trying to keep the enterprise system up and running for transactional purposes. With SAS, you only need to ask for the raw data.

Project example

Planning for the new library

The NCSU Libraries is in the process of moving nearly 1.5 million items to a new library with both open shelving and an automatic retrieval system. One challenge has been to accurately identify and count items as moving to the new library or remaining in their current location. This process is complicated by the fact that we have a set of very detailed criteria to determine if any particular item will be moved. Once identified as moving to the new library, we also need to distinguish whether or not an item will be on open shelves or in the automatic retrieval system.

Using SAS, we are able to precisely code every one of 3,303,566 items as moving or staying. We can then analyze and report on different facets of the collection and use the information to adjust our move criteria if needed.

The first step in this project is to extract raw, delimited data from our ILS. To do so, we worked with our ILS administrator to regularly output two files. One file includes item level data and the other includes title level data. The common key variable/field between the two files is a Title Control number with a one-to-many relationship between titles and items. By requesting raw ILS data, our ILS administrator only needed to write a single query to pull the data rather than attempt to accommodate move criteria that changed and evolved during the planning process.

Table 1 shows the data fields that were extracted from our ILS at the unique item level. Table 2 shows the data fields that were extracted at the unique title level. The common joining key between the two tables is the Titlecontrol. Keeping the title and author data separate from the item level data speeds processing significantly. Titles and author data can then be joined with item level data for output of title lists.

Table 1

Alphabetic List of Variables for the ITEM level data					
Variable	**Type**	**Len**	**Format**	**Label**	**Notes**
callnum	Char	55		Call Number	
currentloc	Char	15		Current Location	
dateinventoried	Num	8	DATETIME	Date Inventoried	
homeloc	Char	15		Home Location	
inhousecharges	Num	8		In House Charges	
itemcreationdate	Num	8	DATETIME	Item Creation Date	
itemid	Char	20		Item ID	Unique ID for each item
itemtype	Char	15		Item Type	
lastcharged	Num	8	DATETIME	Last Charged Date	
library	Char	10		Library	
scheme	Char	12		Class Scheme	
sortcall	Char	110		Shelving Key	Formatted, sortable call number
timesinventoried	Num	8		Times Inventoried	

Alphabetic List of Variables for the ITEM level data					
Variable	**Type**	**Len**	**Format**	**Label**	**Notes**
titlecontrol	Char	20		Title Control	Foreign key to match to title level bibliographic data
totalcharges	Num	8		Total Charges	

Table 2

Alphabetic List of Variables for the TITLE level data				
Variable	**Type**	**Len**	**Label**	**Notes**
author	Char	400	Author	
pubyear	Num	4	Publication Year	
title	Char	1000	Title	
titlecontrol	Char	25	Title Control	Foreign key to match to item level bibliographic data

The second step is to flag each item in the item level table to indicate its move status and the criteria or reason for the item's status. To do so, we create two new variables or fields on the item level table: "Hunt Status" and "Criteria". These new variables are populated based on a range of factors. For example, items falling within certain call number ranges are flagged to be moved to the new library while items from certain branch locations or of certain item types are flagged to remain in their current locations.

Creating new variables in SAS is a simple process. The following code in Figure 1 creates and labels the flag variables. To populate the new variables, we can use SQL syntax or traditional SAS syntax. Figure 2 shows a portion of the code used to populate the flag variables based on the item's formatted call number.

Figure 1

```
*create the Hunt flag and Hunt criteria variables;
data <NEW DATA SET>;
       length hunt 3 criteria $12;
       label hunt = 'Hunt Status'
              criteria = 'Criteria for Hunt move';
       set <ORIGINAL DATA SET>;
run;
```

Figure 2

```
*update the Hunt flag if in correct call number range;
proc sql;
       update sirsi.items
       set Hunt = 1, criteria = 'callnumber'
       where sortcall between 'HF 005548' and 'HF 005548.69' or
              sortcall between 'Q  000300' and 'Q  000390.99999' or
              sortcall between 'QA 000075' and 'QA 000076.99999' or
              sortcall between 'QA 000801' and 'QA 009999.99999' or

       <… Call number ranges continue …>

              sortcall between 'TS 000001' and 'TS 009999.99999' or
              sortcall between 'TX 000955' and 'TX 001099.99999'    ;
quit;#
```

Some additional code is required but once the two flag variables have been populated, we can precisely report at the item level. In doing so, we can answer a range of questions related to planning the new library. For example, we can precisely identify how many items from each location will be moved to the new library. We can also run growth projections for the new library based on the identified set of items to be moved.

Figure 3 shows a portion of an excel file generated from a SAS reporting procedure to identify and count each item's move status. Circled in red is an example of how by using SAS, we can identify even singular items.

Figure 3

A	B	C	D

Number of items to be moved to HUNT Library

Items from SATELLITE

		Hunt Status	
Criteria for Hunt move	**Item Type**	**No**	**Yes**
		38,127	***735,798***
No criteria for move	MANUSCRIPT	10	.
	MAP	24,118	.
No criteria for move		*24,128*	.
Withdrawn Design items	AUD-CD	3	.
	BOOK	8,484	.
	CD-ROM	128	.
	SERIAL	4,716	.
	SOFTWARE	1	.
	SRLCIRC	607	.
	VID-CASS	53	.
	VID-DVD	7	.
Withdrawn Design items		*13,999*	.

Building a Consolidated Usage Statistics Analysis Tool

Darby Orcutt and Genya O'Gara previously led a full monographic use study at The NCSU Libraries in 2009, covering a decade's worth of circulation data ("Looking Forward by Looking Back: Books at the End of the Books," *Proceedings of the XXX Charleston Conference,* 2010, 117-124). That project had immediate application for altering our book approval plan and collection managers' selecting habits in light of large budget cuts, but also gave us a benchmark for future studies of the use of monographs at a time when we were poised to begin adding e-books in a major way. When Libraries Fellow Brian Norberg joined us in July 2010, he also found tremendous value in that study as a way to get to know the collections in his assigned subject areas, and he and I began to discuss how to not only bring the usage study forward in a dynamic way, but to integrate circulation data with other sorts of use data, beginning with inter-library borrowing data—and, in future, e-book usage data. The consolidated statistics analysis tool will allow collection managers to have a window not just into the use of our owned collections, but into our patrons' use of information that we provide via multiple channels.

In teaching library school courses on collection development, Darby Orcutt regularly refers to what he terms the "concentric circles" of ownership and access. The work of collection management, especially in this hybrid to digital era and in the context of our necessary consortial collecting, demands a broad view of not only what content is owned, but also what is made available to our users, and making appropriate decisions as to where certain content needs to fall.

Database Design

The usage stat analysis tool is a Ruby on Rails application. A MySQL database drives the application and contains three tables: the primary Resources table and associated ILLs and ILSs tables. The Resources table has information common to both ILS and ILL data, like title, author, and call number.

Field	Type	Null	Key
id	int(11)	NO	PRI
uniq_id	varchar(255)	YES	
title	varchar(255)	YES	
author	varchar(255)	YES	
citation_date	varchar(255)	YES	
call_number	varchar(255)	YES	
item_type	varchar(255)	YES	
created_at	varchar(255)	YES	
updated_at	datetime	YES	
record_type	varchar(255)	YES	
creation_date	varchar(255)	YES	
uses	int(11)	YES	
location	varchar(255)	YES	
sortable_call_number	varchar(255)	YES	

The schema for the Resources table.

A resource can have one of two associated records, which contain information specific to each data set. If the data comes from the integrated library system, an ILS record will be created that stores details about a title's location, library identification number, and total charges.

Field	Type	Null	Key
id	int(11)	NO	PRI
resource_id	int(11)	NO	
barcode	varchar(255)	YES	
flexkey	varchar(255)	YES	
library	varchar(255)	YES	
create_date	date	YES	
total_charges	int(11)	YES	
last_charged	date	YES	
last_discharged	date	YES	
renewals	int(11)	YES	
inhouse_charges	int(11)	YES	
times_inventoried	int(11)	YES	
date_inventoried	date	YES	
created_at	varchar(255)	YES	
updated_at	datetime	YES	

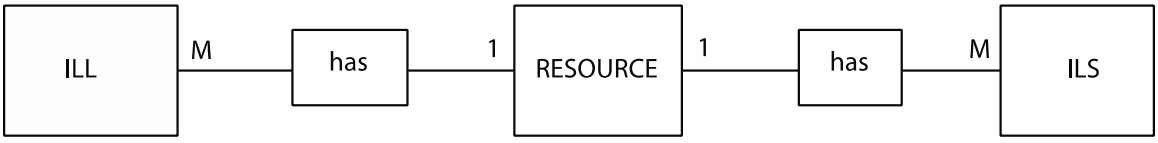
The schema for the ILS table.

An ILL record is created when the data comes from Illiad and contains information about the patron requesting the title and where and when the request occurred.

Field	Type	Null	Key
id	int(11)	NO	PRI
resource_id	int(11)	NO	
identification_number	varchar(255)	YES	
old_id	varchar(255)	YES	
imprint	varchar(255)	YES	
language	varchar(255)	YES	
month_requested	varchar(255)	YES	
year_requested	varchar(255)	YES	
patron_status	varchar(255)	YES	
patron_dept	varchar(255)	YES	
lending_library	varchar(255)	YES	
created_at	varchar(255)	YES	
updated_at	datetime	YES	
request_status	varchar(255)	YES	
dewey	varchar(255)	YES	

The schema for the ILL table.
#

Each ILS and ILL record contains a resource id field that associates it with a Resource record. A Resource record can have many ILL and ILS records, but each ILL and ILS record can only have one Resource record. These associations ensure a user can search ILS and ILL data together or separately.



Entity-Relationship Model for the database.

How Data Is Loaded: The Rake Tasks

Data is entered and updated in the application using the Ruby build program, Rake. Rake is a command line task management tool. In this application, Rake is utilized to clean and organize the ILL and ILS data, query the database for existing records, and add resources if they don't already exist in the database. There are two Rake tasks that can be run in the application.

The first Rake task loads the data. In order to run the task, the user opens a terminal window and types "rake FILE_PATH=/path/to/file.txt db:load_all_data" to begin the load process. What records are created depends on the name of the file being run. For this reason, files must be placed in a certain directory and given a certain name. An example of a filename may be "ILL_book_May-June2011.txt". The Rake task will split the filename and look for "books", "serials", or "sirsi" to determine how the data is to be managed. The text file is parsed line by line, using a Ruby library called FasterCSV. FasterCSV opens the comma-separated text file and turns each line into an array of its values.

During the load, the rake task retrieves each value in the array and stores that value in the correct field of a Resource record and associated ILL or ILS record. But before any records are generated, the Rake task searches the database for barcode, ISSN, or ISBN of each line to establish whether a record already exists in the Resources table. How the Rake task knows what unique identifier to search is again based on the filename. If none of the aforementioned unique identifiers occur in a line of data, the Rake task knows to look in other values to deter-

mine uniqueness and to search the Resources table for a pre-existing record.

When a Resource record with the line's unique identifier does exist, the Rake task skips creating a new Resource record and queries the ILL or ILS tables. These tables are searched to establish if a record exists that contains the same data as the line currently being loaded. If a line comes from ILL data, the Rake task queries the ILLiad id stored in the ILL table. ILL data is extracted from an Access database that merges periodic Illiad reports so all records have an id number. The Rake task pulls out this id number for each record, adds a "b" or "s" in front of it depending on whether it is a serial or book ILL, and places the new identifier in the old_id column for the record in the ILLs table. (In the future, the Libraries will explore pulling data directly from Illiad and using the unique identifiers assigned in its software to eliminate the extra step needed to collect ILL stats.)

This old_id is then used to ensure matching records of an ILL transaction are not recorded in the ILL table by accidentally loading the same data twice. If a line's old_old already appears in the table, no record is created and a message is printed on the command line telling the user about the duplication. Otherwise, a new ILL record is created. The Rake task has a similar feature when adding records to the ILS table. It queries the barcode field in the table to see if it matches the barcode of the incoming line. When the barcodes are different a new record is generated and associated with a Resource record. In both the ILL and ILS tables, there is a resource_id field that is populated with the id number of the Resource record queried or added in the first part of the Rake task when a new record is created. A record is inserted into the ILS table only if the barcode is unique or the total charges of an existing record has changed. Hence, there will be several ILS records for resources that circulate, much like there will be multiple ILL records for resources that have been ILLed more than once. This feature of the Rake task allows the user to track changes in in circulation and ILL activity over time. How regularly data is loaded into the database determines how granular and useful the statistics one can get from the application.

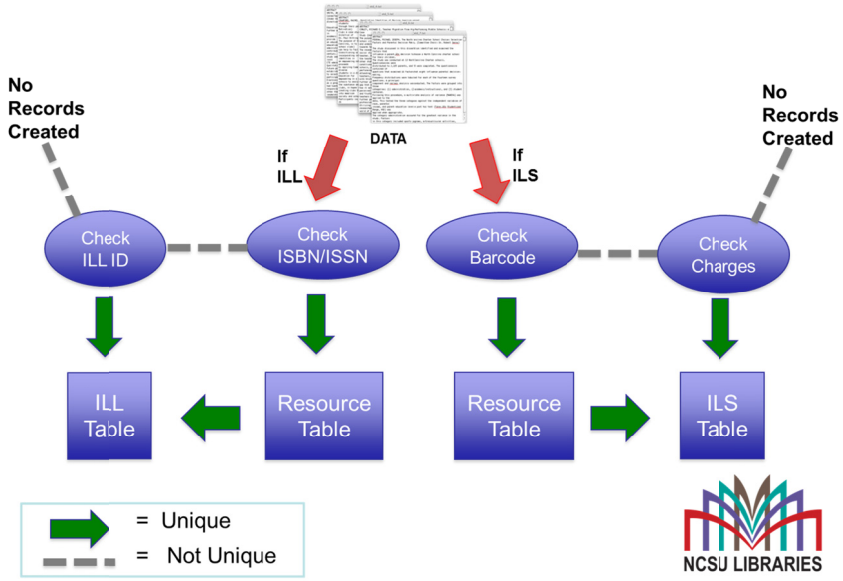
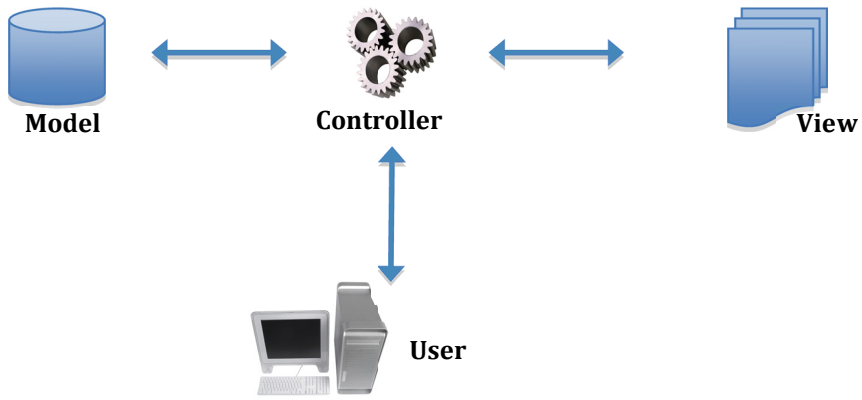


Image of the load process for the Usage Stat Tool.

The other Rake task in the application updates ILL use so that it can be tracked like catalog circulation. Again, the user opens a terminal window, but for this task types, "rake db:update_uses", to start the process. The task queries the Resources table, looking for resources with associated ILL records. It then counts the number of ILLs associated with a resource and stores that number in the uses column of the Resource record. As the task is running, it will print the id number of the resource record it is cur-

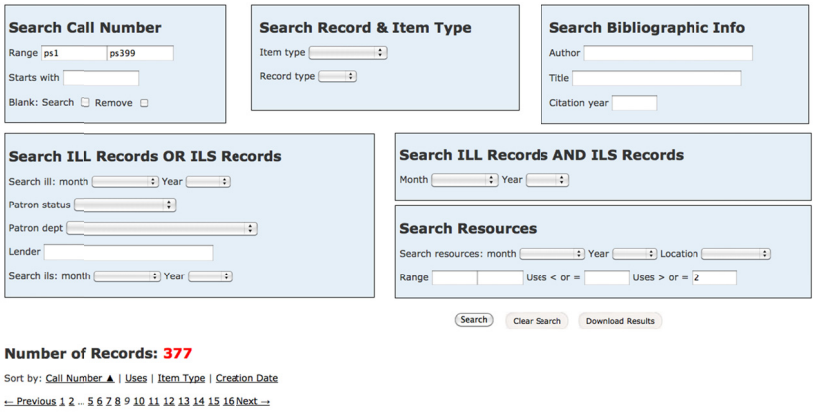
rently processing and the number of ILLs for that record on the command line. This task should be run every time new data is loaded in the application to ensure the database is up to date.

How the Application Works



Rails uses an MVC framework to interact with a database. In the MVC framework, the user sends the server a request. The controller grabs the request, sends it to the Model, which queries the database and send the query results back to the Controller. The Controller than send the results to the view to be processed and styled for the browser. Finally, the manipulated results are sent back to the controller and on to the user.

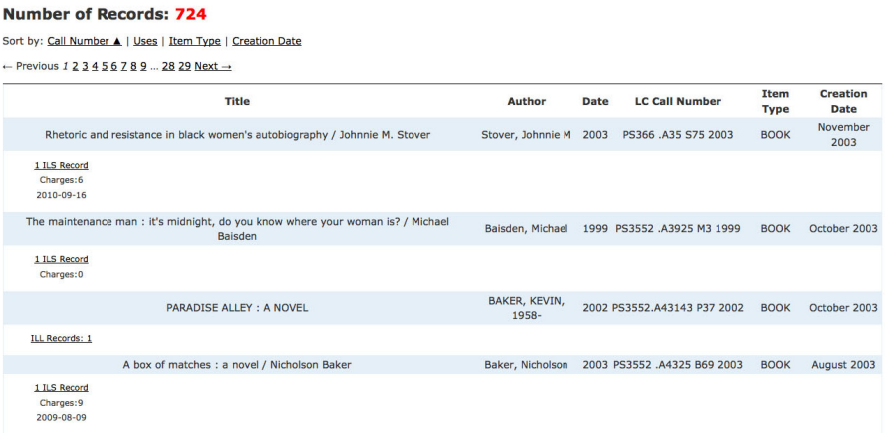
This application contains a Ruby library called Metasearch, which uses the Rails' MVC framework to provide an interface for searching the data through the browser. Metasearch allows for the creation of text boxes and dropdown menus in the View of the application that can be used to send requests to the database. When a text box or dropdown menu is created, it is associated with a certain field and table in the database and a type of query (i.e., contains, equals, starts with, greater than, less than). The user types into a text box or selects a value from a dropdown menu and it gets sent along to the Controller in a hash that stores that input. The Controller processes the hash and sends a request to the Model asking for records that match the input, and the Model creates an SQL query to talk to the database. After the Model retrieves records from the database, it sends the data to the Controller, which stores the data as an array and determines how it will be served up to the View. In the View is where the array gets iterated through one record at a time and displayed to the browser.



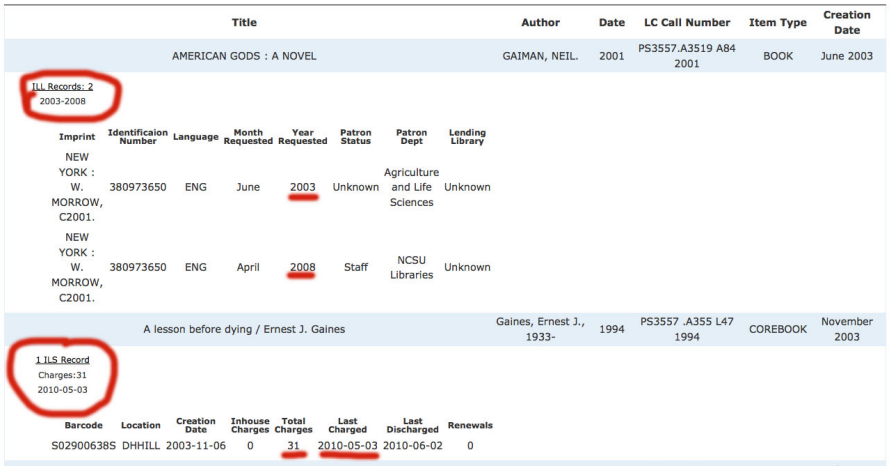
View of the search interface for the Usage Stat Tool.

When a user loads the application, the first 25 records from the Resources table appear in a table below a series of search boxes, along with the total number of resources and the ability to paginate through and sort these records by call number, uses, item type, or creation date. The records appear in ascending order by call number. Checking the "Remove" box in the "Search Call Number" box in the top, left-hand side of the screen will get rid of any resources that have a blank call number.

In the table, the title, author, citation date, call number, item type, and creation date of each Resource record is displayed in a light blue background. Below this information is a link that tells the number of ILS or ILL records associated with the resource. If a resource has an associated ILS record, the total charges of the resource and, if the total charges are not zero, the date of the last charge appear under the link. When a resource has more than one associated ILL record, the dates of the first and last ILL activity for the resource are shown. Clicking the link will expose more thorough information about the ILL or ILS records.



View of the search results returned in the browser.



Title	Author	Date	LC Call Number	Item Type	Creation Date
AMERICAN GODS : A NOVEL	GAIMAN, NEIL.	2001	PS3557.A3519 A84 2001	BOOK	June 2003

ILL Records: 2
2003-2008

Imprint	Identificaion Number	Language	Month Requested	Year Requested	Patron Status	Patron Dept	Lending Library
NEW YORK : W. MORROW, C2001.	380973650	ENG	June	2003	Unknown	Agriculture and Life Sciences	Unknown
NEW YORK : W. MORROW, C2001.	380973650	ENG	April	2008	Staff	NCSU Libraries	Unknown

Title	Author	Date	LC Call Number	Item Type	Creation Date
A lesson before dying / Ernest J. Gaines	Gaines, Ernest J., 1933-	1994	PS3557 .A355 L47 1994	COREBOOK	November 2003

1 ILS Record
Charges:31
2010-05-03

Barcode	Location	Creation Date	Inhouse Charges	Total Charges	Last Charged	Last Discharged	Renewals
S02900638S	DHHILL	2003-11-06	0	31	2010-05-03	2010-06-02	0

View of the associated ILL and ILS records when the link is opened.

A user can search the ILL/ILS data by typing in any of the various text boxes or selecting from the dropdown menus. Some of the search categories are call number, bibliographic information, item type, and uses. The application can query the Resources table, which is a combination of the ILL and ILS data, or it can search the ILL and ILS material separately. Two searches this application can perform that the Libraries current systems for storing ILL and ILS cannot are searches for number of uses and across call number ranges. Search results can be viewed on the browser and paginated through 25 records at a time, or they can be downloaded and looked at in a program like Excel.

What the Database Does

At this early stage in its development, this database:

- Replaces our old, Excel-based ILL monthly reports (saving staff time and effort, as well as compiling total ILL activity, which is much more valuable in decision-making than time-limited/snapshot data)
- Offers side by side presentation of ILL and circulation data (enabling quick and dirty analysis at a glance)
- Provides easy means of producing basic data sets for analysis on the fly (e.g., high use or low use items within particular call number ranges, etc.)
- Places reporting/analysis/collection profile functions within the capacity of all selectors to quickly compile for themselves
- Can be relatively easily built upon as needs arise.

Challenges and Future Directions

One of the challenges getting ILL and ILS data to talk to one another is the difference in the way each is collected. Because ILL records are not part of the Libraries' catalog, they have no barcodes. Similarly, Sirsi does not record ISBN/ISSN numbers for each item in the catalog. Instead, Sirsi stores an array of all possible ISBN/ISSN for an item at the title control level. For these reasons, it is very difficult to associate both ILL and ILS records with a resource. This problem arises when a book that the Libraries owns is ILLed because all its copies are being circulated. The possibility of storing the Sirsi array as a string in the database that can be searched for the existence of an incoming ISBN/ISSN is being explored. Currently, a user can see all ILL or ILS record for a certain resource together because records are sorted by call number.

Other future developments on the usage stat analysis tool include making the data load and retrieval process more efficient, adding the ability to search the data by fund code, and using Google Chart to provide a visualization feature.

Electronic Theses and Dissertations Reference Extractor

Purpose

This tool fills a large gap in our data, especially when facing serials review. We have had use data, on the one hand—including use data that has become much more refined and standard over the years—but this is still a fairly raw measure, and certainly does not indicate the type of user (e.g., un-

dergraduate, graduate, faculty, other), which can be crucial to a selection and de-selection decision. We have also had Library Journal Usage Reports (LJUR) data, but these are also incomplete and imprecise measures that generally omit much of the use that we would consider significant. ETD reference data supplies in many ways the "missing link" between faculty publication and citation. Use by Ph.D. students may be the single most crucial factor in collection decisions, but it's one we (and most if not all of our peers) have had to extrapolate from other measures or sampling studies, compile from laborious citation analysis, or simply guess. This current project serves as proof of concept that this data can be relatively painlessly obtained.

How the Application Works

The ETD Reference Extractor is only in its first stages of development, and has been implemented on a limited scale. Currently, the tool uses two command line scripts to retrieve Education ETDs from the NCSU Libraries Digital Repository, extract them, and pull out only the reference section in each PDF.

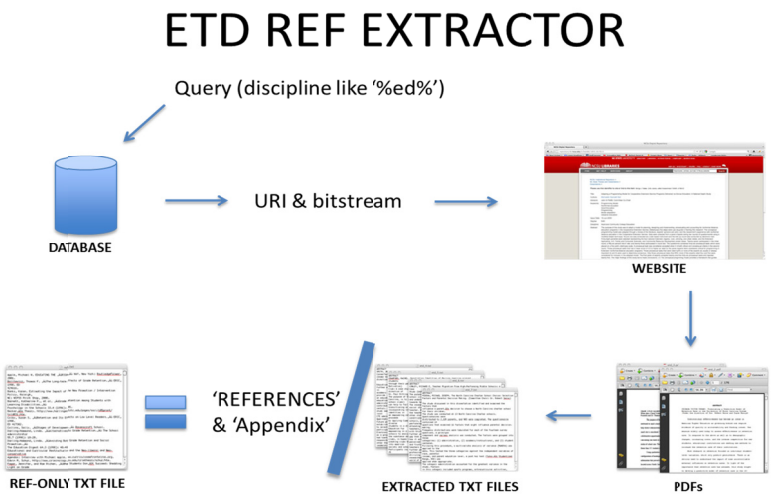


Image of the how the ETD Extractor works on the back end.

The first script searches the Libraries' Dspace repository for any ETD that matches the parameters of the query entered into the script. The Libraries is working on an interface to search the ETD database; but, as of now, the query has to be hardcoded into the script to search for ETDs by discipline. The results of the query are stored in an array that gets iterated through to find the handle and bit stream for the PDF associated with each record. This information is used to recreate the URL where each PDF can be downloaded. The Ruby library, Net/HTTP is then utilized to visit each URL and retrieve its PDF. The contents of each PDF are then written into a new file on the local machine and placed in the same file folder.

Once the PDF files are on the local machine, the second script is run. This script utilizes the Ruby library, DocSplit, to extract the text out of the downloaded PDFs. DocSplit requires that GraphicsMagick, an image processing software, be loaded on the machine in order to do the extraction. To run the script, the user opens a terminal window and types, "Ruby pdf_extractor.rb file_path = /path/to/pdfs". The script then gathers all the files in the file_path folder into a variable that is passed to the DocSplit extract_text method. Text files are created for each PDF in the file_path folder and added to a new folder. Each new text file is iterated through and searched for certain words.

Currently, the tool has only been tested on ETDs from the College of Education. For theses and dis-

sertations submitted in this college, the most common format style for a work cited is to introduce source material with the title, "REFERENCES". It is also common for submissions to include appendices to the document directly after the references with a title, "Appendix". There are a few exceptions to these usual submission practices and for those outliers, the script subs out whatever word or phrase occurs in the document with the standard word. The script then splits each extracted PDF on "REFERENCES" and puts these parts into an array. "REFERENCES" typically occurs only twice in a document (once in the Table of Contents and once introducing the work cited) so to find the part of the document that contains the source material you can select the last part of the array and store it in a variable. This variable stores only the part from the work cited to the end of the document. By splitting the variable on "Appendix" and grapping the first part of the array, the script pulls out only the source material. Finally, the script creates a new text document and adds only the source material of each PDF into it.

Visualizing the Data

Currently, Collection Management is using an open access, web-based visualization tool called, Voyeur, to analyze the new reference-only document. Voyeur allows the user to upload a text document and search for any occurrence of a word or phrase. When a search is done, Voyeur returns all variations found within the document that match the query and how many times and where each variation occurs. A few searches that are possible with Voyeur are: how often a journal title is used throughout the theses and dissertations in a given department and how often journals in general or a certain publisher is cited.

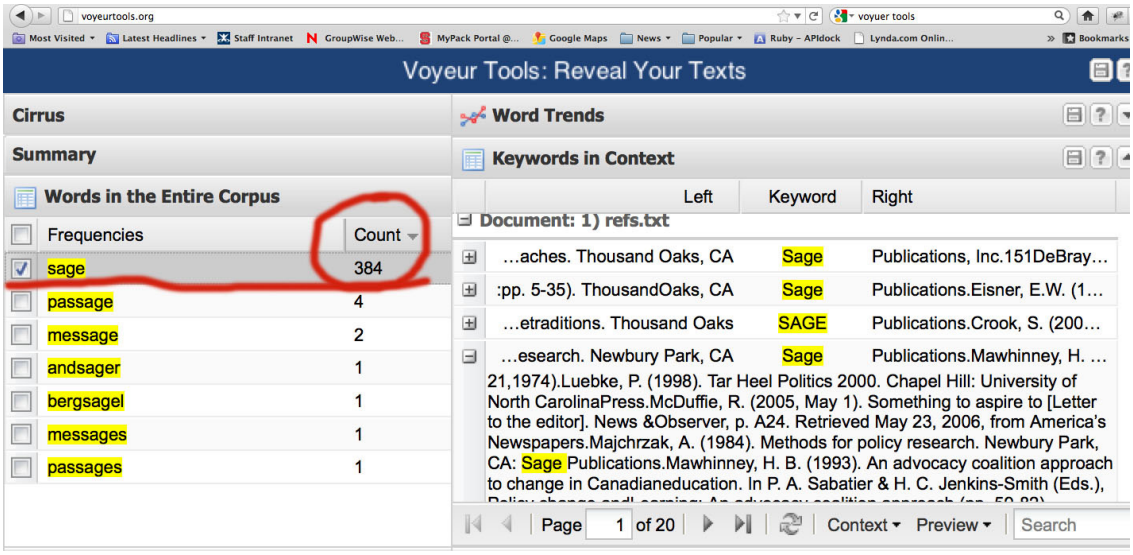


Image of searching for the number of times Sage is cited in the ETDs. Voyeur Tools, Stéfan Sinclair & Geoffrey Rockwell (©2011)#

Benefits and Future Developments

Among other things, the ETD Reference Extractor will:

- Provide easily compiled data regarding graduate student citation of journals and other information resources
- Obviate any need for traditional, labor-intensive citation analyses
- Allow librarians to regularly and holistically compare graduate student citation patterns with those of faculty, as well as with use statistics from other sources

- Serve as a means of gathering snapshot and longitudinal data in order to better understand trends in particular programs, interdisciplinary use of resources, and other yet-to-be-determined questions of interest for library research and informed practice
- Hopefully prove a valuable tool that we can make available for use by the wider library community in conducting their own local and eventually inter-institutional analyses.

There are several planned enhancements to the Electronic Theses and Dissertations Reference Extractor application. The first enhancement is to convert the Python script into Ruby and tie the two scripts together so that multiple steps are not needed to run the application. Once the two scripts are combined, the next development is to make the application more scalable. A graphical user interface will be created to allow users to select ETDs at a more granular level by offering the ability to search for year ranges and departments.

This advanced selection feature will facilitate changes in how the application extracts the references from the downloaded PDFs, as well as how NCSU Libraries' Collection Management department collects information on department ETD submission policies. In order to pull ETDs by department and year accurately, Collection Management must keep track of any changes in submission policy regarding departmental naming. Similarly, work needs to be done in Collection Management to document variations among departments in submission practices relating to work cited and appendices in order to train the application to extract references from more ETDs.

These and many other successful data analysis efforts have come through longstanding commitments the Collection Management Department of the NCSU Libraries to intentionally cultivate an entrepreneurial culture with regard to data tools. Collection managers developed the requisite computing skills for each of the above projects while on the job, through both a formal class and self-teaching. The team-oriented environment of the department further served to encourage and push librarians beyond initially perceived limits. The products and rewards of this emphasis on learning and developing new data tools are many, and our model should certainly not be beyond the reach of institutions who value data-rich decision-making.